Diagnosis of Tidal Turbine Vibration Data through Deep Neural Networks

Grant S. Galloway, Victoria M. Catterson[2], Thomas Fay[3], Andrew Robb[4] and Craig Love[5]

[1, 2] *Institute for Energy and Environment, Dept. EEE, University of Strathclyde, Glasgow, G1 1XW, UK*
grant.galloway.2013@uni.strath.ac.uk
v.m.catterson@strath.ac.uk

[3, 4, 5] *Andritz Hydro Hammerfest, Glasgow, G52 4RU, UK*

ABSTRACT

Tidal power is an emerging field of renewable energy, harnessing the power of regular and predictable tidal currents. However, maintenance of submerged equipment is a major challenge. Routine visual inspections of equipment must be performed onshore, requiring the costly removal of turbines from the sea bed and resulting in long periods of downtime. The development of condition monitoring techniques providing automated fault detection can therefore be extremely beneficial to this industry, reducing the dependency on inspections and allowing maintenance to be planned efficiently.

This paper investigates the use of deep learning approaches for fault detection within a tidal turbine's generator from vibration data. Training and testing data were recorded over two deployment periods of operation from an accelerometer sensor placed within the nacelle of the turbine, representing ideal and faulty responses over a range of operating conditions. The paper evaluates a deep learning approach through a stacked autoencoder network in comparison to feature-based classification methods, where features have been extracted over varying rotation speeds through the Vold-Kalman filter.

1. INTRODUCTION

Traditional methods of fault diagnostics for rotating machinery typically involve a feature extraction stage, where signal processing techniques are used to extract representations of raw data more meaningful of specific faults or failure modes. This usually involves implementing a series of signal processing techniques to identify or isolate specific features of the raw data, followed by a classification stage to distinguish healthy data from faulty responses, figure 1.

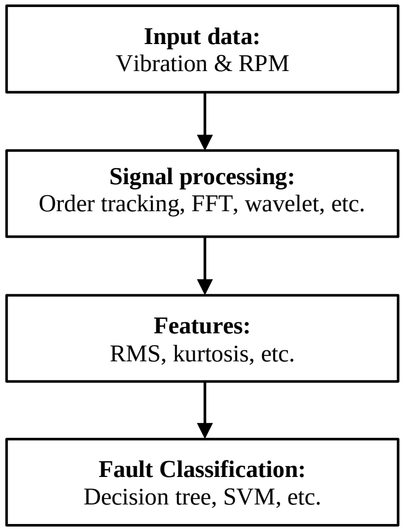


Figure 1. Generic high-level feature-based diagnostics process

To date, many feature extraction techniques have been studied for calculating features most representative of faults within vibration data, including various time domain and frequency domain methods to detect both stationary and non-stationary fault signatures. A detailed review of different fault diagnostics techniques for condition monitoring and preventative maintenance using audio and vibration signals is provided by Henríquez, Alonso, Ferrer and Travieso (2014).

However, recent advances in the field of machine learning have led to implementation of 'deep learning' approaches, where feature engineering can replaced by unsupervised learning through neural network architectures (Schmidhuber, 2015).

The aim of this paper is to explore the use of deep learning applied for diagnosing a generator fault within a tidal turbine through vibration data. A deep neural network, consisting of stacked sparse autoencoders and a softmax classifier, is used to learn features from vibration data and perform diagnosis through classification. This deep learning approach is compared to a feature-based method, where features linked to specific failure modes are extracted from vibration data using the Vold-Kalman filter and classified using decision trees, support vector machines and K-nearest neighbours.

The paper finds that deep learning through a network of stacked sparse autoencoders can offer improved diagnostic performance in comparison to feature-based methods. This method can be particularly useful for applications in which there is limited operational data (e.g. rotation speed measurement) or for applications where fault dynamics are not well enough understood for engineering appropriate features.

2. TIDAL TURBINE DATA

Data available for this study was provided by Andritz Hydro Hammerfest, a manufacturer of tidal turbines. Condition data was sourced from the HS1000, a prototype commercial scale 1 MW tidal turbine in operation off the coast of Orkney, Scotland at a tidal array test site as part of the European Marine Energy Centre (EMEC).

The HS1000 turbine has an open-bladed horizontal axis design and is fixed in position to the seabed, as shown in figure 2. Three blades drive a 1 MW rated induction generator through a gearbox, converting tidal flow rates to generator rotation speeds exceeding 1000 RPM.

Figure 2. Andritz Hydro Hammerfest HS1000 Tidal Turbine

2.1. Accelerometer Data

Vibrations were recorded through a tri-axial accelerometer sensor fixed to the nacelle of the turbine, adjacent to the generator and sampled at 2 kHz. Available datasets detailed a number of vibration measurements over two separate deployments of the turbine. Measurements were taken during a range of different operating and weather conditions, where the turbine experienced constantly variable loading and rotation speed due to changing tidal flow rates and turbulence.

Variation in rotor speed can make frequency-based analysis of vibration signals challenging, where frequency components relating to rotations of equipment vary in time. Order analysis techniques are often used to compensate for variations in rotor speed through techniques such as computed resampling or the Vold-Kalman filter (Wang & Heyns, 2011).

2.2. Generator Fault Case

During the turbine's second deployment, the turbine displayed a change in response causing increased vibrations within the turbine's generator. This can be observed in figure 3 showing the average spectrum of vibration data for each deployment. Computed resampling (Wang & Heynes, 2011) has been used to normalize the frequency spectrum to represent rotational orders of the high speed shaft (HSS).

A clear increase in vibration from the generator's poles can be observed at the 6th order of the HSS. High vibrations were also identified at 100 Hz (a harmonic of the line frequency) and at harmonics of the generator rotation frequency within the turbine at different loads. The turbine is currently awaiting decommissioning following the completion of its testing program.

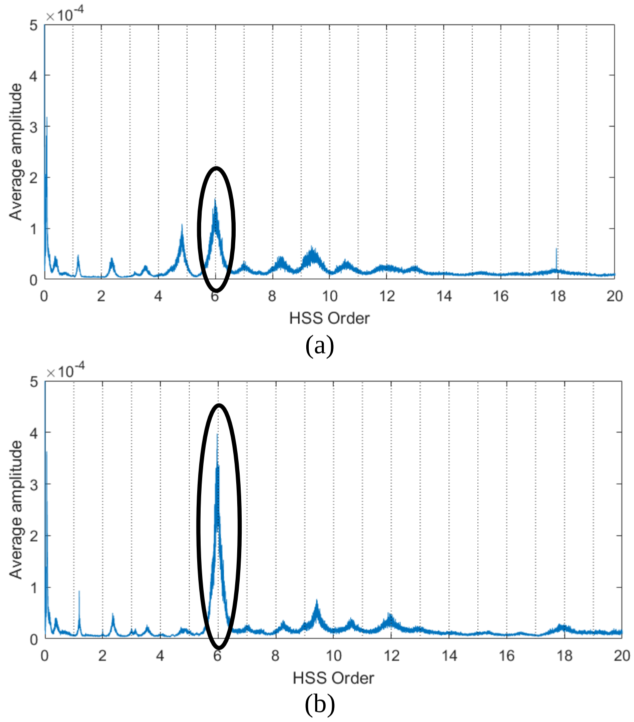


Figure 3. Average frequency response of (a) healthy tidal turbine and (b) generator fault

3. METHODOLOGY

This paper aims to compare the performance of feature-based classification to deep learning for the diagnosis of a generator fault within a tidal turbine.

Vibration features were extracted through the Vold-Kalman filter, isolating a set of specific frequency components during variable rotor speeds. This feature set was then used to train a number of classification techniques including decision trees, support vector machines and k-nearest neighbors. These results were used as a benchmark for the deep learning approach.

Deep learning was performed through a deep neural network, consisting of layers of sparse autoencoders and a softmax classifier. Sparse autoencoders learned features from spectrograms of raw vibration data and were used to train a connected softmax classifier layer. The classification performance was then improved by retraining the network as a whole through a process known as ‘fine-tuning’. These steps are described in greater detail below.

3.1. Feature-Based Classification

Vibration components characteristic of generator faults were extracted from vibration data through the Vold-Kalman filter. A feature-set containing the root mean square (RMS) of select frequency components was then used to train a number of classifiers including support vector machine (SVM), decision trees and k-nearest neighbours. Each classifier was trained to classify healthy and faulty behaviour of the generator.

Generator faults produce excess vibration at harmonics of the generator rotation speed, caused by unevenly distributed electromagnetic forces or mechanical wear (Scheffer & Girdhar, 2004). These fault signals are generally stationary under steady operating conditions, where the distribution of the signal does not change over time. This differs from non-stationary fault signals, generally observed from faults within rotating components such as gearboxes or bearings causing periodic impacts, where the distribution of signals change over time.

The Vold-Kalman filter is a suitable feature extraction method for this fault case as it is able to isolate multiple components from a vibration signal who’s fundamental frequencies change because of variable rotation speed, but are otherwise stationary signals.

3.1.1. Vold-Kalman Filter

The Vold-Kalman filter acts as a moving band-pass filter, set to track sine waves with high slew rates (Vold & Leuridan, 1995). This was used as a form of order tracking, extracting features from raw vibration data recorded during high variation in rotor speed.

Given an RPM measurement and an order to be extracted, this technique consists of solving a set of linear least squares equations known as the structural and data equations (Vold & Leuridan, 1995). A second generation Vold-Kalman filter (Tume, 2005) was implemented in MATLAB, using an implementation by van der Seijs (2012).

The following components were taken as features for vibrations over each axis, detailing vibrations associated with generator faults (Scheffer & Girdhar, 2004):

- Generator shaft rotation frequency and 1st harmonic
- Generator poles vibration frequency and 1st harmonic
- Line frequency 1st harmonic (100 Hz)

The root mean square (RMS) values of each extracted frequency component were then used as features for classification, calculated as in Eq. (1) where x_i is the amplitude of a frequency component over window length N.

$$x_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2} \tag{1}$$

3.1.2. Classification Techniques

A number of classification techniques were compared to build a benchmark for feature-based diagnostics. These techniques included the support vector machine (SVM), decision tree and K-nearest neighbours (KNN)

Support Vector Machine (SVM)

Support vector machines (SVM) classify data through the definition of a hyperplane, separating data points of different classes in the feature space. This hyperplane is defined to maximize the distance between the hyperplane and the nearest data point (the margin), giving optimum separation between classes. Support vectors are data points closest to the hyperplane and define the boundaries between classes, figure 4. A detailed explanation of support vector machines is provided by Cortes and Vapnik (1995).

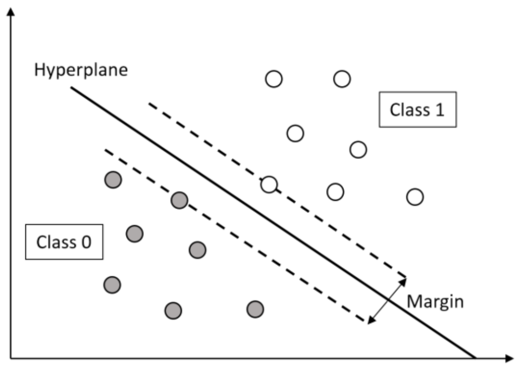


Figure 4. Support Vector Machine classification

Support vector machines are natively linear, but can be made non-linear through the 'kernel trick': representing input data in a higher dimensional space through a kernel function (Shawe-Taylor & Cristianini, 2004). In this study, three forms of SVM classifier were used: linear, quadratic, and cubic.

Decision Tree

A decision tree is a structure containing a set of decision rules that predict a set of outcomes (or class values for classification) from input data (Rivest, 1987). Decision trees consist of nodes representing a series of decisions and outcomes for input data. Internal nodes have 'branches' leading to subsequent nodes, performing decisions on the input data. 'Leaves' are nodes with no further decision branches and represent the most appropriate class for the input data leading to that point. Figure 5 represents a simple decision tree for input data x_i with targets y.

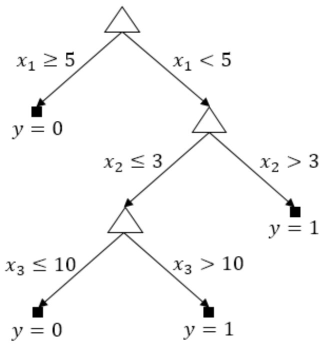


Figure 5. Decision Tree classification

k-Nearest Neighbors

The k-nearest neighbors method performs classification by measuring the distance between test data and training data. A class is assigned to test data by selecting the majority class of the k training data points closest to the test data (Altman, 1992).

The Euclidean distance metric was used to find the k closest data points, Eq. (2), where $p = (p_1, p_2, \dots, p_n)$ and $q = (q_1, q_2, \dots, q_n)$ are Cartesian coordinates of features in n-dimenional space.

$$\mathrm{d}(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{2}$$

3.2. Deep Learning through Sparse Autoencoders

Deep learning is a form of machine learning where multi-layered architectures are used to autonomously learn features and classify data, replacing feature extraction. These architectures are commonly based upon neural networks through techniques such as autoencoders, deep neural networks, deep belief networks and convolutional neural networks (Schmidhuber, 2015).

Deep learning has currently been applied to various condition monitoring and machinery diagnostic applications, for example:

- Sparse coding to extract features for bearing fault detection (Liu, Liu & Huang, 2011)
- Stacked autoencoders to detect valve and bearing faults from acoustic data (Verma, Gupta, Sharma & Sevakula, 2013)
- Deep neural networks for diagnosing partial discharge data in high voltage assets (Catterson & Sheng, 2015)
- Deep neural networks for diagnosing bearing and planetary gearbox faults (Jia, Lei, Lin, Zhou & Lu, 2016)
- Stacked autoencoders for fault diagnosis of bearings based on the wavelet transform (Junbo, Weining, Junfeng & Xueqian, 2015)
- Sparse coding to learn the response of bearing failure (Martin del Campo & Sandin, 2015)
- Stacked autoencoders with support vector regression for health state estimation in fuel cell systems (Qiao & Xun, 2015)

In this paper, deep learning is performed though a network of stacked autoencoders, where sparse autoencoder layers learn features from vibration data and are stacked on top of a softmax classification layer. The network was trained on spectrograms generated from raw vibration data, inspired by (Lee, Largman, Pham & Ng, 2009) where spectrogram data was used to train a convolutional neural network for speech recognition.

3.2.1. Stacked Autoencoder Deep Neural Networks

Autoencoders are a form of neural network used for unsupervised learning (Vincent, Larochelle, Lajoie, Bengio & Manzagol, 2010). Their aim is to replicate an input x as its output $\hat{x}$ through a number of hidden neurons (figure 6a). The network learns the function $h_{W,b}(x)$ to best approximate the input, Eq. (3).

Each hidden neuron in the autoencoder encodes input data x to $a^{(1)}$, known as neuron's activation. This is detailed in Eq. (4) where f is a transfer function (commonly sigmoid), $W^{(1)}$ is a weight matrix and $b^{(1)}$ is bias vector.

This $a^{(1)}$ is then decoded in the following layer through transfer function f and parameters $W^{(2)}$ and $b^{(2)}$ to best estimate the original input vector x as in Eq. (5).

By limiting the number of hidden neurons, the network is forced to learn a compressed representation of the input data through parameters W and b. Hidden neurons can therefore be seen as features that have been learned by the network,

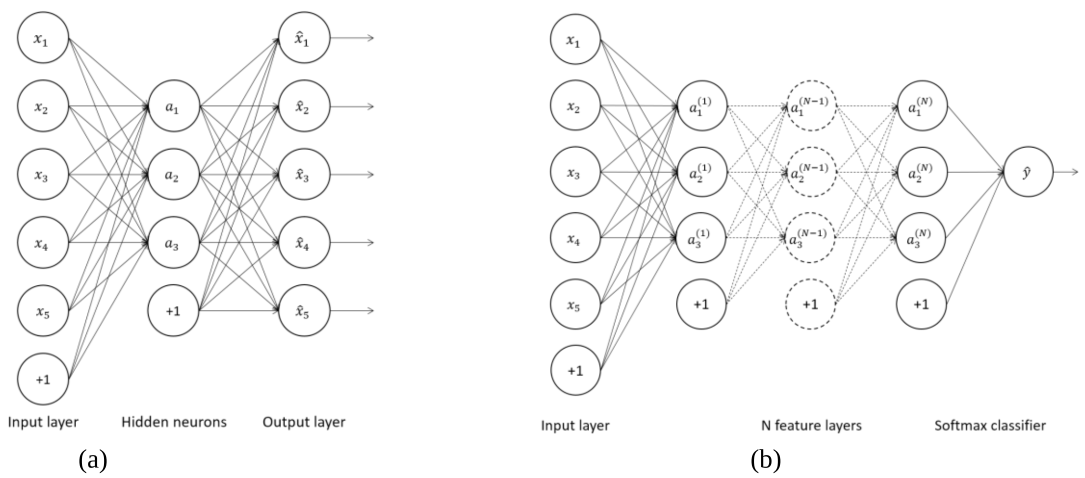


Figure 6. Network configurations for (a) a sparse autoencoder and (b) a stacked autoencoder deep neural network

representing the structure of input data. This can form representations similar to principle component analysis (PCA), however it is a non-linear process.

$$h_{W,b}(x) \approx x \tag{3}$$

$$a^{(1)} = f\left(W^{(1)}x + b^{(1)}\right) \tag{4}$$

$$\hat{x} = f\left(W^{(2)}a^{(1)} + b^{(2)}\right) \tag{5}$$

A deep neural network can be created through stacking multiple autoencoder layers on top of a final classification layer. This is shown in figure 6b, where N autoencoders are connected together before a softmax classification layer. Stacking multiple autoencoder layers together allows the network to learn higher order features, where each successive layer represents additional complexity within the input data.

3.2.2. Training the Network

The network is trained using gradient descent, where the aim is to minimise a cost function $J(W,b)$. This cost function is defined as in Eq. (6). The first term is a mean squared error term, where K is the number of examples/observations, $x^{(k)}$ is the input data at each example/observation, $y^{(k)}$ is the output data at each example/observation and $h_{W,b}$ is the function performed by each neuron dependent on weight W and bias b parameters.

The second term, $\Omega_{weights}$, is a weight decay term (also called L2 regularization) used to improve generalization and prevent overfitting. Third term, $\Omega_{sparsity}$, is a term to encourage more sparse activations of hidden neurons (Olshausen & Field, 1997). Parameters λ and β control the influence of the weight decay and sparsity terms respectively.

$$J(W,b) = \left[\frac{1}{K}\sum_{k=1}^{K}\frac{1}{2}\left\|h_{W,b}\left(x^{(k)}\right) - y^{(k)}\right\|^2\right] + \lambda\Omega_{weights} + \beta\Omega_{\text{sparsity}} \tag{6}$$

Derivatives of the cost function required for the gradient descent technique are computed through back propagation. In this study this was performed through the scaled conjugate gradient algorithm, explained in detail by (Møller, 1993).

The network is first pre-trained, where each autoencoder layer is trained separately (Hinton, Osindero & Teh, 2006). The first layer autoencoder is trained on raw input data $x^{(k)}$ to learn feature parameters $W^{(1)}$ and $b^{(1)}$. Activations $a^{(1)}$ are then used as inputs when training the next autoencoder layer, learning feature parameters $W^{(2)}$ and $b^{(2)}$ and activations $a^{(2)}$. This is repeated for N autoencoder layers where the final set of activations $a^{(N)}$ are used to train the softmax classifier with target outputs y.

Following pre-training, the network is then 'fine-tuned' to improve the classification performance (LeCun, Bengio & Hinton, 2015). This step allows the network to improve the features learned during unsupervised pre-training to maximize margins between classification classes. During fine-tuning, all layers are trained together through back propagation using the same cost function as in Eq. (6).

4. RESULTS

In this study, a deep learning approach was compared to a feature-based approach for the diagnosis of a tidal turbine generator fault.

The feature-based diagnostic method used a feature set consisting of the RMS of select frequency components known to indicate generator faults, extracted from vibration data using the Vold-Kalman filter. This feature set was then used to train classifiers including support vector machine

(SVM), decision tree and K-nearest neighbours (KNN) to detect when the turbine's generator was operating under healthy and faulty conditions.

Deep learning was then performed through a stacked autoencoder network. The network was trained on spectrograms generated from raw vibration data and learned feature representations of input data through its hidden layers. The network's final layer then used to classify both healthy and faulty generator behaviour.

In each case, the classification accuracy was measured as a percentage as in Eq. (7), where $N_{correct\ diagnoses}$ refers to the number of correctly classified test examples and $N_{test\ examples}$ is the total number of test examples.

$$accuracy\ (\%) = \left(\frac{N_{correct\ diagnoses}}{N_{test\ examples}}\right) \times 100 \qquad (7)$$

4.1. Feature-based Diagnostics

The following frequency components were extracted from X, Y and Z axis vibration data using the Vold-Kalman filter, based on the measured rotation speed of the high speed shaft:

- Generator shaft rotation frequency and 1st harmonic
- Generator poles vibration frequency and 1st harmonic
- Line frequency 1st harmonic (100 Hz)

These frequency components detailed specific vibration components from the turbine's generator, linked to generator faults (Scheffer & Girdhar, 2004).

A feature set containing the RMS of each frequency component over a 1 second window was used as the input to each binary classifier. To improve classification accuracy, features were scaled using the Z-score standardization method (Kreyszig, 1979), Eq. (8) where x is feature data, $\bar{x}$ is the mean and σ is the standard deviation .

$$x' = \frac{x - \bar{x}}{\sigma} \qquad (8)$$

Feature data from the two deployments were used to train binary classification methods capable of identifying this particular fault within the generator for automated diagnosis. Data from the turbine's first deployment were used as healthy (class 0) and data from the second deployment, during periods where excessive vibrations were observed, were used as faulty data (class 1). 70% of data was randomly select for training models, with the remaining 30% used for testing.

Table 1 details these results, where a cubic SVM was found to give the highest classification accuracy of 96.86%.

Table 1: Feature-based classification accuracies

Classification Method		**Testing accuracy**
SVM	Linear SVM	93.01 %
	Quadratic SVM	96.06 %
	Cubic SVM	**96.86 %**
Decision Tree	20 maximum nodes	95.96 %
	50 maximum nodes	96.85 %
	100 maximum nodes	96.39 %
KNN	$k = 1$	96.42 %
	$k = 10$	96.76 %
	$k = 100$	95.59 %

4.2. Stacked Autoencoder Deep Neural Network

A stacked autoencoder deep neural network was trained using spectrograms generated from raw vibration data as inputs. A series of autoencoders were first trained in an unsupervised fashion to learn features from the spectrogram data. These features were then fed into a softmax classifier layer to categorise healthy behaviour and a generator fault. The network was then retrained through back propagation in a process called 'fine-tuning' to adjust the features learned to improve the classification accuracy.

4.2.1. Spectrogram Data

Narrow spectrograms, referred to as spectrogram 'slices', were used as inputs to the deep neural network. Spectrogram slices were generated from raw vibration data through the Short Time Fourier Transform (STFT) (Sejdić, Djurović & Jiang, 2009). 10 STFTs were performed on vibration data every 0.5 seconds with an overlapping window of length 1 second, figure 7. The frequency scale was truncated to 500 Hz to reduce dimensionality. Spectrogram slices (10x500 sized vectors) gave a time-frequency representation of the vibration data, allowing the network to potentially learn representations of both stationary and non-stationary signals. 70% of data was randomly select for training the network, with the remaining 30% used for testing.

4.2.2. Classification Results

Networks were tested with two configurations: a single autoencoder layer and two autoencoder layers. Increasing the number of autoencoder layers increases the complexity of features learned by the network. The number of hidden units in each autoencoder layer was also altered, changing the number of features learned by each layer.

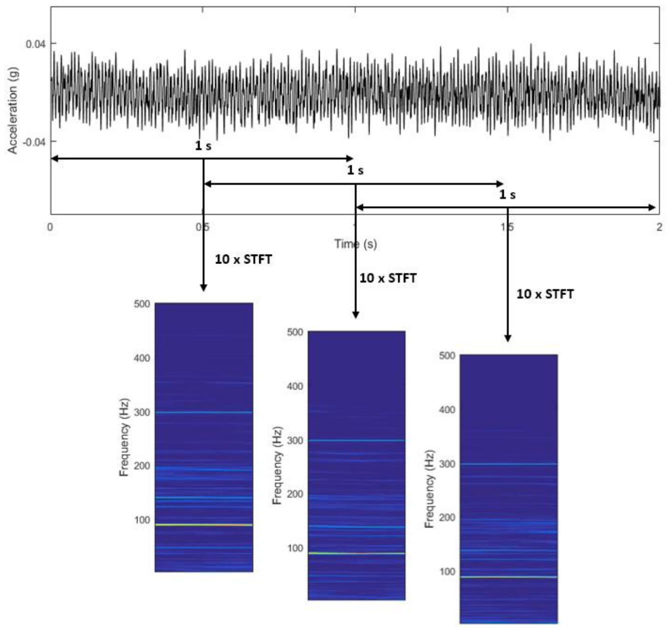


Figure 7. Sampling spectrogram 'slices' from raw vibration data

The sparsity proportion parameter β and weight decay parameter λ in the cost function, Eq. (6), were set to 0.05 and 0.001 respectively.

The autoencoder activation function h, detailed in Eq. (4), was chosen as a rectified linear unit, Eq. (9). This function is popular for deep networks and tends to lead to sparse activation of hidden units (LeCun, et al., 2015).

$$f(x) = \max(0, x) \tag{9}$$

Table 2 details the classification accuracies for each network configuration, where the network was tested after pre-training and fine-tuning stages. Results show the deep learning approach offers improved classification accuracy over a feature-based method of fault diagnosis.

Classification accuracy after the pre-training stage increased as the number of hidden units was increased. This shows that learning more features gives a better representation of the turbine's vibration behaviour for both healthy and faulty cases. However, accuracy increased to 100% after fine-tuning for any number of hidden units. This reveals the strength of deep learning approaches, as pre-trained features can be further optimised to reduce the classification error. Increasing the number of autoencoder layers offers no significant improvement in classification accuracy and in fact appears to reduce the classification accuracy of pre-trained features in some cases.

4.2.3. Feature Visualisation

Despite the neural network being primarily a 'black-box' technique, features learned by autoencoder layers can be visualised by examining the weights W of each hidden unit, Eq. (10).

$$x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{N}\left(W_{ij}^{(1)}\right)^2}} \tag{10}$$

Figure 8 displays x_j for the 5 hidden units in the first autoencoder layer with the highest activation for both healthy and faulty conditions. Under healthy conditions features contain a strong low frequency component, relating to normal structural oscillations of the turbine's nacelle and support structure. Given data under faulty conditions the network appears to have learned a series of fault signatures, the most prominent relating to the generator poles' vibration at different rotation speeds.

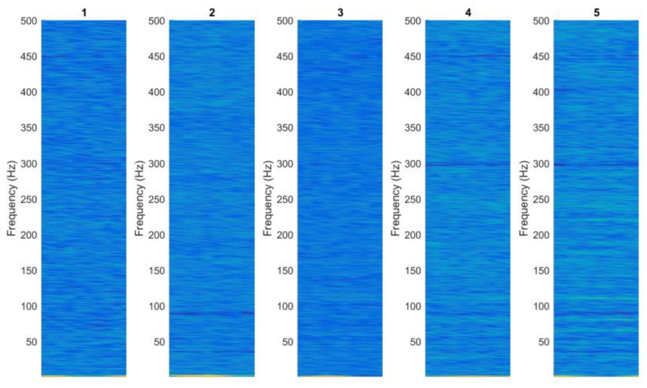
Table 2**:** Stacked autoencoder network accuracy

Number of autoencoder layers	**Number of Hidden Units**	**Pre-training accuracy**	**Fine-tuning accuracy**
1	5	66.77 %	100 %
	10	80.82 %	100 %
	20	96.65 %	100 %
	50	98.05 %	100 %
	100	**99.53 %**	**100 %**
2	5	67.91 %	100 %
	10	77.24 %	100 %
	30	97.62 %	100 %
	50	97.89 %	100 %
	100	**99.93 %**	**100 %**

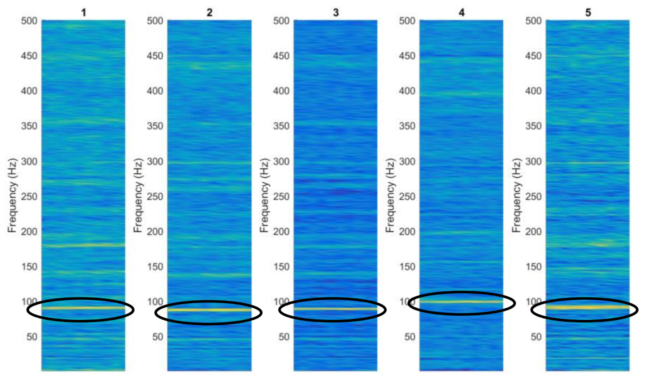
5. DISCUSSION

This study has shown deep learning approaches can have a number of advantages over feature-based classification methods. Firstly, the deep learning method implemented in this study has shown a classification accuracy of 100%. This is even the case for networks with very few learned features after fine-tuning.

In addition, such high classification accuracy was achieved from raw vibration data without supporting data describing the operating conditions or loading on the turbine (such as rotor speed or tidal flow rates). Visualisation of the weights of hidden neurons for faulty data showed that the network learned features that were indicative of fault signatures under different loading conditions. Little engineering knowledge of the system (under both healthy and faulty conditions) is required to achieve good diagnostic performance, provided



(a)



(b)

Figure 8. Features learned by stacked autoencoder network for (a) healthy and (b) faulty vibrations

there is access to sufficient data. This may be beneficial for new or increasingly complex systems, where the dynamics of faults are not well understood, or for applications where the specifications and/or loading of machinery are unknown.

However in comparison to feature-based methods, deep learning methods may have shortcomings. For example, sufficient labelled data is required for classification and diagnosis. In addition, the computational requirements can increase dramatically for larger datasets.

6. CONCLUSION

In this study, results have shown a deep learning approach can improve upon classification accuracy of feature-based methods for diagnosing a fault within a tidal turbine's generator from vibration data.

Feature-based classification methods were tested, extracting features from vibration data using the Vold-Kalman filter. These features were then used to train SVM, decision tree and KNN classifiers. The highest classification accuracy from this method was 96.86%, obtained through a cubic SVM.

A deep neural network of stacked autoencoders was then trained with spectrograms constructed from raw vibration data, learning the response of the tidal turbine under variable loading conditions and identifying a fault within the turbine's generator. The network achieved classification accuracy of 100%, improving upon the accuracy of feature-based methods, without the additional loading data (such as rotor speed).

Future work will involve benchmarking deep learning approaches against other common feature extraction methods for additional vibration datasets, testing the method's ability to detect faults in other rotating machine components (such as gearboxes and bearings), where fault signals may be non-stationary and more complex.

REFERENCES

Altman, N. S. (1992). An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *The American Statistician*, 46(3), pp. 175-185. doi:10.1080/00031305.1992.10475879

Catterson, V. M., & Sheng, B. (2015). Deep Neural Networks for Understanding and Diagnosing Partial Discharge Data. *IEEE Electrical Insulation Conference* (pp. 218-221). June 7-10, Seattle, WA. doi: 10.1109/ICACACT.2014.7223616

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20, pp. 273-297. doi: 10.1023/A:1022627411411

Henríquez, P., Alonso, J. B., Ferrer, M. A., & Travieso, C. M. (2014). Review of Automatic Fault Diagnosis Systems Using Audio and Vibration Signals. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 44(5), pp. 642-652. doi: 10.1109/TSMCC.2013.2257752

Hinton, G. E., Osindero, S. & Teh, Y. W. (2006). A fast learning algorithm for deep belief nets. *Neural Comp*, 18, pp. 1527-1554. doi: 10.1162/neco.2006.18.7.1527

Jia, F., Lei, Y., Lin, J., Zhou, X., & Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 73, pp. 303-315. doi: 10.1016/j.ymssp.2015.10.025

Junbo, T., Weining, L., Junfeng, T., & Xueqian, W. (2015). Fault Diagnosis Method Study in Roller Bearing Based on Wavelet Transform and Stacked Auto-encoder. Qingdao: *27th Chinese Control and Decision Conference (CCDC)* (pp. 4608-4613), May 23-25, Qingdao. doi: 10.1109/CCDC.2015.7162738

Kreyszig, E. (1979). *Advanced Engineering Mathematics* (4th ed.). New York, NY: John Wiley & Sons.

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep Learning. *Nature*, 521, pp. 436-444. doi: 10.1038/nature14539

Lee, H., Largman, Y., Pham, P., & Ng, A. Y. (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. *Advances in Neural Information Processing Systems*, 22, pp. 1096-1104.

Liu, H., Liu, C., & Huang, Y. (2011). Adaptive feature extraction using sparse coding for machinery fault diagnosis. *Mechanical Systems and Signal Processing*, 25(2), pp. 558-574. doi: 10.1016/j.ymssp.2010.07.019

Martin del Campo, S., & Sandin, F. (2015). Towards zero-configuration condition monitoring based on dictionary learning. *23rd European Signal Processing Conference (EUSIPCO)* (pp. 1306-1310). August 31 – September 4, Nice, France.

Møller, M. F. (1993). A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6(4), pp. 525-533. doi: 10.1016/S0893-6080(05)80056-5

Olshausen, B. A., & Field, D. J. (1997). Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1? *Vision Research*, 37(23), pp. 3311-3325. doi: 10.1016/S0042-6989(97)00169-7

Qiao, L. Q., & Xun, L. J. (2015). State of health estimation combining robust deep feature learning with support vector regression. *34th Chinese Control Conference (CCC)* (pp. 6207-6212). July 28-30, Hangzhou, China. doi: 10.1109/ChiCC.2015.7260613

Rivest, R. L. (1987). Learning Decision Lists. *Machine Learning*, 2(3), pp. 229-246. doi: 10.1023/A:1022607331053

Rokach, L., & Maimon, O. (2005). Top-Down Induction of Decision Trees Classifiers - A Survey. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 35(4), pp. 476-487. doi: 10.1109/TSMCC.2004.843247

Scheffer, C., & Girdhar, P. (2004). *Practical Machinery Vibration Analysis & Predictive Maintenance*. Oxford, UK: Newnes.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, pp. 85-117. doi: 10.1016/j.neunet.2014.09.003

Sejdić, E., Djurović, I. & Jiang, J. (2009). Time-frequency feature representation using energy concentration: An overview of recent advances. *Digital Signal Processing*, 19(1), pp. 153-183. doi: 10.1016/j.dsp.2007.12.004

Shawe-Taylor, J., & Cristianini, N. (2004*). Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge University Press.

Tuma, J. (2005). Setting the Passband Width in the Vold-Kalman Order Tracking Filter. *12th International Congress on Sound and Vibration (ICSV12)* (pp. 1-8). July 11-14, Lisbon.

Verma, N. K., Gupta, V. K., Sharma, M., & Sevakula, R. K. (2013). Intelligent Condition Based Monitoring of Rotating Machines using Sparse Auto-encoders. *Proceedings of IEEE Conference on Prognostics and Health Management* (pp. 1-7). June 24-27, Gaithersburg, MD. doi: 10.1109/ICPHM.2013.6621447

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. *Journal of Machine Learning Research*, 11, pp. 3371-3408.

van der Seijs, M. (2013) Second generation Vold-Kalman Order Filtering. http://www.mathworks.com/matlabcentral/fileexchange/36277-second-generation-vold-kalman-order-filtering

Vold, H., & Leuridan, J. (1995). High Resolution Order Tracking at Extreme Slew Rates Using Kalman Tracking Filters. *Shock and Vibration*, 2(6), pp. 507-515. doi: 10.4271/931288

Wang, K., & Heyns, P. S. (17-19 June 2011). A Comparison between Two Conventional Order Tracking Techniques in Rotating Machine Diagnostics. *IEEE International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE)* (pp. 478-481). June 17-19, Xi'an. doi: 10.1109/ICQR2MSE.2011.5976657

BIOGRAPHIES

Grant S. Galloway is a PhD student within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his M.Eng in Electronic and Electrical Engineering from the University of Strathclyde in 2013. His PhD focuses on condition monitoring and prognostics for tidal turbines, in collaboration with Andritz Hydro Hammerfest, a leading tidal turbine manufacturer.

Victoria M. Catterson is a Lecturer within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. She received her B.Eng. (Hons) and Ph.D. degrees from the University of Strathclyde in 2003 and 2007 respectively. Her research interests include condition monitoring, diagnostics, and prognostics for power engineering applications.

Thomas Fay is a Control and Instrumentation Engineer with Andritz Hydro Hammerfest, Glasgow, UK.

Andrew Robb is a Mechanical Engineer with Andritz Hydro Hammerfest, Glasgow, UK.

Craig Love is a Turbine System Engineer with Andritz Hydro Hammerfest, Glasgow, UK.